Proofs as schedules

Emmanuel Beffara\textsuperscript{1} \quad Virgile Mogbil\textsuperscript{2}

\textsuperscript{1}IML, CNRS & Université d’Aix-Marseille
\textsuperscript{2}LIPN, CNRS & Université Paris Nord

Chocola – 14 March 2013

EB and VM. \textit{Proofs as executions}.
Plan

Introduction
  Concurrency and linear logic
  ...and why we should search further

Pairings and determinisation
  Schedules of processes

Logic for schedules
  Typing schedules
  Soundness and completeness

What next
  CPS for processes
  Boxless versions
The *formulae as types* approach:

- formula $\leftrightarrow$ type
- proof rules $\leftrightarrow$ primitive instructions
- proof $\leftrightarrow$ program
- normalization $\leftrightarrow$ evaluation

The *proof search* approach:

- formula $\leftrightarrow$ program
- proof rules $\leftrightarrow$ operational semantics
- proof $\leftrightarrow$ successful run
Typing the $\pi$-calculus in linear logic

Typing rules

Axiom and cut:

$$u \xrightarrow{\circ} v \vdash u : \downarrow A^\perp, v : \uparrow A$$

$$P \vdash \Gamma, \bar{x} : A \quad Q \vdash \bar{x} : A^\perp, \Delta$$

$$(\nu \bar{x})(P \mid Q) \vdash \Gamma, \Delta$$

Multiplicatives:

$$P \vdash \Gamma, \bar{x} : A \quad Q \vdash \bar{y} : B, \Delta$$

$$P \mid Q \vdash \Gamma, \bar{x}\bar{y} : A \otimes B, \Delta$$

$$P \vdash \Gamma, \bar{x} : A, \bar{y} : B$$

$$P \vdash \Gamma, \bar{x}\bar{y} : A \otimes B$$

Actions:

$$P \vdash \Gamma, \bar{x} : A$$

$$u(\bar{x}).P \vdash \Gamma, u : \downarrow A$$

$$P \vdash \Gamma, \bar{x} : A$$

$$\bar{u}(\bar{x}).P \vdash \Gamma, u : \uparrow A$$

E. Beffara, V. Mogbil (IML, LIPN)
Typing the $\pi$-calculus in linear logic

Properties of the system

Good things:

- Typed processes cannot diverge or deadlock.
- Typing is preserved by reduction (up to structural congruence).
- Explains translations of the $\lambda$-calculus into the $\pi$-calculus.
- Extends to differential LL.
Typing the $\pi$-calculus in linear logic

Properties of the system

Good things:

- Typed processes cannot diverge or deadlock.
- Typing is preserved by reduction (up to structural congruence).
- Explains translations of the $\lambda$-calculus into the $\pi$-calculus.
- Extends to differential LL.

Shortcomings:

- Typed processes are confluent.
- Many well-behaved interaction patterns are not typable.

$\vdash a.\overline{b} | b.\overline{c} | \overline{a}.c.d$
A few observations

Proof normalization, aka *cut elimination*:
- the meaning of a proof is in its normal form,
- normalization is an *explicitation* procedure,
- it really wants to be confluent.

Interpretation of concurrent processes:
- the meaning is the *interaction*, the final (irreducible) state is less relevant,
- a given process may behave very differently depending on scheduling decisions.
Proofs as schedules

The principles of our new interpretation:

- formula $\leftrightarrow$ type of interaction
- proof rules $\leftrightarrow$ primitives for building schedules
- proof $\leftrightarrow$ schedule for a program
- normalization $\leftrightarrow$ evaluation

What this is not:

- **Curry-Howard** for processes:
  proofs are not programs, but behaviours of programs
- **Proof search:**
  the dynamics is not in proof construction but in cut-elimination
- **Specification, verification:**
  only “may”-style properties can be expressed, currently
We consider a CCS-style process calculus.

\[
P, Q := 1 \quad \text{inaction} \\
a.P \quad \text{perform } a \text{ then do } P \\
P \mid Q \quad \text{interaction of } P \text{ and } Q \\
(\nu a)P \quad \text{scope restriction}
\]

There is one source of non-determinism:
the pairing of associated events upon synchronization

\[
a.P \mid a.Q \mid \bar{a}.R \rightarrow \begin{cases} 
a.P \mid Q \mid R \\
P \mid a.Q \mid R \end{cases}
\]
Pairings

Definition

A *pairing* is an association between occurrences of dual actions

\[ p_1 : \quad P = a.b.A \mid \bar{a}.c.B \mid \bar{b}.\bar{c}.C \mid a.\bar{c} \]

\[ p_2 : \quad P = a.b.A \mid \bar{a}.c.B \mid \bar{b}.\bar{c}.C \mid a.\bar{c} \]

Definition

A *determinisation* of \( P \) along a pairing \( p \) is a renaming \( \partial_p(P) \) of actions in \( P \) where names are equal only for related actions.

\[ \partial_{p_1}(P) = a'.b'.\partial(A) \mid \bar{a}.c.\partial(B) \mid \bar{b}''.\bar{c}''.\partial(C) \mid a.\bar{c} \]

\[ \partial_{p_2}(P) = a.b.\partial(A) \mid \bar{a}.c.\partial(B) \mid \bar{b}.\bar{c}.\partial(C) \mid a'.\bar{c}' \]
Facts about pairings:

- each run induces a pairing
- runs are equivalent up to permutation of independent events iff they induce the same pairing
- if $p$ is a consistent pairing of $P$ then $p$ is the unique maximal consistent pairing of $\partial_p(P)$

Hence pairings are *execution schedules* and determinized terms represent them inside the process language.

Logic will type these schedules.
A logic of schedules

The language

Types of schedules:

\[ A, B := \langle a \rangle A \quad \text{do action } a \text{ and then act as } A \]

\[ A \otimes B \quad \text{two independent parts, one as } A, \text{ the other as } B \]

\[ A \uplus B \quad A \text{ and } B \text{ are both exhibited, but correlated} \]

\[ \alpha \quad \text{an unspecified behaviour} \]

\[ \alpha^\perp \quad \text{something that can interact with } \alpha \]

Transforming schedules:

\[ A_1, ..., A_n \vdash B \quad \text{behave as type } B \text{ using one schedule of each type } A_i \]
A logic of schedules

Typing rules

Axiom and cut:

\[
\begin{align*}
1 & \vdash \alpha^\perp, \alpha \\
\hline
\Gamma, \Delta & \vdash A \\
\hline
P & \vdash \Gamma, A, B \\
Q & \vdash A^\perp, \Delta \\
\Gamma, \Delta & \vdash P \mid Q
\end{align*}
\]

Multiplicatives:

\[
\begin{align*}
P & \vdash \Gamma, A \\
Q & \vdash B, \Delta \\
\Gamma, A \otimes B, \Delta & \vdash P \mid Q \\
\Gamma, A \boxtimes B & \vdash P \vdash \Gamma, A, B
\end{align*}
\]

Actions:

\[
\begin{align*}
P & \vdash \Gamma, A \\
\Gamma, \langle a \rangle A & \vdash a.P
\end{align*}
\]
The role of the axiom rule
Two-sided presentation

\[
\begin{align*}
1 & : \alpha \vdash \alpha \\
\overline{b} & : \alpha \vdash \langle \overline{b} \rangle \alpha \\
\overline{a} \overline{b} & : \alpha \vdash \langle \overline{a} \overline{b} \rangle \alpha \\
\overline{c} & : \alpha \vdash \langle \overline{c} \rangle \alpha \\
b \overline{c} & : \langle \overline{b} \rangle \alpha \vdash \langle \overline{c} \rangle \alpha \\
c \overline{d} & : \langle \overline{c} \rangle \alpha \vdash \langle \overline{d} \rangle \alpha \\
\overline{a} \overline{c} \overline{d} & : \langle \overline{a} \overline{c} \overline{d} \rangle \alpha \vdash \langle \overline{d} \rangle \alpha
\end{align*}
\]
The role of the axiom rule

Two-sided presentation

\[
\begin{align*}
1 : \alpha & \vdash \alpha \\
\overline{b} : \alpha & \vdash \langle \overline{b}\rangle \alpha \\
a.\overline{b} : \alpha & \vdash \langle a\overline{b}\rangle \alpha \\
\end{align*}
\]
The role of the axiom rule

Two-sided presentation

\[
\begin{align*}
1 & : \alpha \vdash \alpha \\
\bar{b} & : \alpha \vdash \langle \bar{b} \rangle \alpha \\
a.\bar{b} & : \alpha \vdash \langle a\bar{b} \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
1 & : \alpha \vdash \alpha \\
\bar{c} & : \alpha \vdash \langle \bar{c} \rangle \alpha \\
b.\bar{c} & : \langle \bar{b} \rangle \alpha \vdash \langle \bar{c} \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
1 & : \alpha \vdash \alpha \\
d & : \alpha \vdash \langle d \rangle \alpha \\
c.\bar{d} & : \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
b.\bar{c} & \vdash \langle \bar{b} \rangle \alpha \rightarrow \langle \bar{c} \rangle \alpha \\
c.d & : \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
1 & : \langle \bar{b} \rangle \alpha \vdash \langle \bar{b} \rangle \alpha \\
c.d & : \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
a.\bar{c}.d & : \langle a\bar{b} \rangle \alpha, \langle \bar{b} \rangle \alpha \rightarrow \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
\bar{a}.c.d & : \langle a\bar{b} \rangle \alpha \vdash \langle d \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
a.\bar{b} \mid b.\bar{c} \mid \bar{a}.c.d : \alpha \vdash \langle d \rangle \alpha \\
\end{align*}
\]
The role of the axiom rule
Two-sided presentation

\[
\begin{align*}
1 & : \alpha \vdash \alpha \\
\bar{b} & : \alpha \vdash \langle \bar{b} \rangle \alpha \\
\hline
a.\bar{b} & : \alpha \vdash \langle a\bar{b} \rangle \alpha
\end{align*}
\]

\[
\begin{align*}
1 & : \alpha \vdash \alpha \\
b.\bar{c} & : \langle \bar{b} \rangle \alpha \vdash \langle \bar{c} \rangle \alpha \\
\hline
b.\bar{c} & \vdash \langle \bar{b} \rangle \alpha \rightarrow \langle \bar{c} \rangle \alpha
\end{align*}
\]

\[
\begin{align*}
1 & : \langle \bar{b} \rangle \alpha \vdash \langle \bar{b} \rangle \alpha \\
c.d & : \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \\
\hline
c.d & : \langle \bar{b} \rangle \alpha, \langle \bar{b} \rangle \alpha \rightarrow \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \\
\bar{a}.c.d & : \langle a\bar{b} \rangle \alpha, \langle \bar{b} \rangle \alpha \rightarrow \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha
\end{align*}
\]

\[
\begin{align*}
\bar{a}.c.d & : \langle a\bar{b} \rangle \alpha \vdash \langle d \rangle \alpha \\
\hline
b.\bar{c} & \vdash \langle a\bar{b} \rangle \alpha \vdash \langle d \rangle \alpha
\end{align*}
\]

\[
\begin{align*}
\bar{a}.c.d & : \alpha \vdash \langle d \rangle \alpha
\end{align*}
\]
The role of the axiom rule

Two-sided presentation

<table>
<thead>
<tr>
<th>1</th>
<th>( \alpha \vdash \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \bar{b} )</td>
<td>( \alpha \vdash \langle \bar{b} \rangle \alpha )</td>
</tr>
<tr>
<td>( a.\bar{b} )</td>
<td>( \alpha \vdash \langle a\bar{b} \rangle \alpha )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>( \alpha \vdash \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \bar{c} )</td>
<td>( \alpha \vdash \langle \bar{c} \rangle \alpha )</td>
</tr>
<tr>
<td>( b.\bar{c} )</td>
<td>( \langle \bar{b} \rangle \alpha \vdash \langle \bar{c} \rangle \alpha )</td>
</tr>
<tr>
<td>( b.\bar{c} \vdash \langle \bar{b} \rangle \alpha \rightarrow \langle \bar{c} \rangle \alpha )</td>
<td></td>
</tr>
</tbody>
</table>

| \( \bar{a}.c.d \) | \( \langle a\bar{b} \rangle \alpha, \langle \bar{b} \rangle \alpha \rightarrow \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \) |

<table>
<thead>
<tr>
<th>1</th>
<th>( \alpha \vdash \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \bar{d} )</td>
<td>( \alpha \vdash \langle \bar{d} \rangle \alpha )</td>
</tr>
<tr>
<td>( c.d )</td>
<td>( \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha )</td>
</tr>
</tbody>
</table>

\( a.\bar{b} \mid b.\bar{c} \mid \bar{a}.c.d : \alpha \vdash \langle d \rangle \alpha \)
The role of the axiom rule

Two-sided presentation

\[
\begin{align*}
1 : \alpha & \vdash \alpha \\
\bar{b} : \alpha & \vdash \langle \bar{b} \rangle \alpha \\
a.\bar{b} : \alpha & \vdash \langle a\bar{b} \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
1 : \langle \bar{b} \rangle \alpha & \vdash \langle \bar{b} \rangle \alpha \\
b.\bar{c} : \langle \bar{b} \rangle \alpha & \vdash \langle \bar{c} \rangle \alpha \\
\bar{a}.c.d : \langle a\bar{b} \rangle \alpha, \langle \bar{b} \rangle \alpha & \vdash \langle d \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
1 : \alpha & \vdash \alpha \\
\bar{c} : \alpha & \vdash \langle \bar{c} \rangle \alpha \\
c.d : \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
1 : \alpha & \vdash \alpha \\
\bar{c} : \alpha & \vdash \langle \bar{c} \rangle \alpha \\
c.d : \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
1 : \langle \bar{b} \rangle \alpha & \vdash \langle \bar{b} \rangle \alpha \\
\bar{c} : \alpha & \vdash \langle \bar{c} \rangle \alpha \\
c.d : \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
1 : \alpha & \vdash \alpha \\
\bar{c} : \alpha & \vdash \langle \bar{c} \rangle \alpha \\
c.d : \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \\
\end{align*}
\]

\[
\begin{align*}
1 : \alpha & \vdash \alpha \\
\bar{c} : \alpha & \vdash \langle \bar{c} \rangle \alpha \\
c.d : \langle \bar{c} \rangle \alpha \vdash \langle d \rangle \alpha \\
\end{align*}
\]
The role of the axiom rule

One-sided presentation

Duality: \((A \otimes B)^\perp = A^\perp \pitchfork B^\perp\), \((\langle a \rangle A)^\perp = \langle \bar{a} \rangle (A^\perp)\).
The role of the axiom rule

Proof net presentation

\[ a \bar{b} \cdot 1 \mid (b \bar{c} \cdot 1 \mid \bar{a} \cdot c \cdot d \cdot 1) \]
The role of the axiom rule

Proof net presentation

\[ a.\bar{b}.1 \mid (b.\bar{c}.1 \mid \bar{a}.c.d.1) \]
The role of the axiom rule

Proof net presentation

\[ \overline{b}.1 \mid (b.\overline{c}.1 \mid c.d.1) \]
The role of the axiom rule

Proof net presentation

$$\bar{b}.1 \mid (b.\bar{c}.1 \mid c.d.1)$$
The role of the axiom rule

Proof net presentation

\[ \overline{b}.1 \mid (b.\overline{c}.1 \mid c.d.1) \]
The role of the axiom rule

Proof net presentation

\[ \overline{b}.1 \mid (b.\overline{c}.1 \mid \underline{c}.\underline{d}.1) \]
The role of the axiom rule

Proof net presentation

\[
\bar{b}.1 \mid (\bar{b}.\bar{c}.1 \mid c.d.1)
\]

E. Beffara, V. Mogbil (IML, LIPN)
The role of the axiom rule

Proof net presentation

\[1 \mid (\tilde{c}.1 \mid c.d.1)\]
The role of the axiom rule

Proof net presentation

\[ \langle \bar{c}\rangle \alpha \quad \alpha^\perp \]

\[ \langle c\rangle \alpha^\perp \]

\[ \langle d\rangle \alpha \]

\[ \bar{c}.1 \mid c.d.1 \]
The role of the axiom rule
Proof net presentation

\(\bar{c}.1 \mid c.d.1\)
The role of the axiom rule

Proof net presentation

\[ \langle d \rangle \alpha \]

1 | d.1
The role of the axiom rule
Proof net presentation
Mandatory theorems

Theorem (Soundness)

Typing is preserved by reduction, head cut-elimination steps correspond to execution steps.

- a typed deterministic term cannot deadlock,
- normalization corresponds to a particular execution.
Theorem (Soundness)

Typing is preserved by reduction, head cut-elimination steps correspond to execution steps.

- a typed deterministic term cannot deadlock,
- normalization corresponds to a particular execution.

Theorem (Completeness)

For every lock-avoiding run $P_1 \rightarrow \ldots \rightarrow P_n$ there are corresponding typings such that $\pi_1 : P_1 \vdash \Gamma \rightarrow \ldots \rightarrow \pi_n : P_n \vdash \Gamma$ is a cut elimination sequence.

- need to define “lock-avoiding”
Summing up

- Full calculus
- Determinisation
- Simple calculus
- Typing
- Multiplicative logic
Summing up

- Full calculus
- More expressive logics
- Determinisation
- Simple calculus
- Multiplicative logic
- Typing
From one typing to the other
Continuation passing style in concurrency?

\[
\begin{align*}
\alpha &\perp \\
\langle a \overline{b} \rangle \alpha &\perp \\
\langle \overline{c} \rangle \alpha &\perp \\
\langle b \rangle \alpha &\perp \\
\langle b \rangle \alpha &\perp \otimes \langle \overline{c} \rangle \alpha \\
\langle \overline{b} \rangle \alpha &\otimes \langle \overline{c} \rangle \alpha \\
\langle d \rangle \alpha &
\end{align*}
\]

\[
a \cdot \overline{b} \cdot 1 \mid (b \cdot \overline{c} \cdot 1 \mid \overline{a} \cdot c \cdot d \cdot 1)
\]
From one typing to the other
Continuation passing style in concurrency?

\[ (\nu a)(a(b)\bar{b}\langle u \rangle.1 | (\nu c)(b(y)\bar{c}\langle y \rangle.1 | \bar{a}\langle b \rangle.c(z)\langle d \rangle.z.1)) \]
It *seems* that all actions can be typed the same way:

\[
\langle a \cdot \rangle A := \forall \alpha ((A \rightarrow \alpha) \rightarrow \langle a \rangle \alpha) = \forall \alpha ((A \otimes \alpha^\perp) \not\Rightarrow \langle a \rangle \alpha)
\]
It *seems* that all actions can be typed the same way:

\[
\langle a \cdot \rangle A := \forall \alpha ((A \rightarrow \alpha) \rightarrow \langle a \rangle \alpha) = \forall \alpha ((A \otimes \alpha^\perp) \otimes \langle a \rangle \alpha)
\]

- Cut expansion is acceptable at the logical level.
It *seems* that all actions can be typed the same way:

\[
\langle a \cdot \rangle A := \forall \alpha ((A \rightarrow \alpha) \rightarrow \langle a \rangle \alpha) = \forall \alpha ((A \otimes \alpha^\perp) \otimes \langle a \rangle \alpha)
\]

- Cut expansion is acceptable at the logical level.
- Moreover, there is no real need for boxes in the confluent world!
Getting out of the box

- It *seems* that all actions can be typed the same way:

\[
\langle a \cdot \rangle A := \forall \alpha ((A \rightarrow \alpha) \rightarrow \langle a \rangle \alpha) = \forall \alpha ((A \otimes \alpha^\perp) \otimes \langle a \rangle \alpha)
\]

- Cut expansion is acceptable at the logical level.

- Moreover, there is no real need for boxes in the confluent world!

→ a boxless calculus, reminiscent of translations of $\pi$ into solos
Conclusion, extensions

Current state of affairs:

- A logical description of scheduling in processes
  - describes how schedules can be safely composed
  - normal forms as basic open schedules

- Explicitation of control flow through processes

- Hints for a new study of causality in processes

Possible extensions:

- Connectives to combine related behaviours:

\[
 t_1.(t_2 + f_2 \mid \bar{t}_0) + f_1.(t_2 \cdot \bar{t}_0 + f_1 \cdot \bar{f}_0) \vdash B[t_1, f_1] \otimes B[t_2, f_2] \twoheadrightarrow B[t_0, f_0]
\]

where \( B[t, f] := \alpha \twoheadrightarrow \langle \bar{t} \rangle \alpha \oplus \langle \bar{f} \rangle \alpha \)

- Predicates to describe states

- Richer action modalities for richer communication
C’est tout pour aujourd’hui.